

## GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	Advanced Topics in Modern Software Engineering / Advanced Topics in Modern Software Engineering	
Ders Kodu / Course Code	9105055562017	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	Second Cycle / Second Cycle	
Ders Akts Kredi / ECTS	8.00	
Haftalık Ders Saati (Kuramsal) / Course Hours For Week (Theoretical)	3.00	
Haftalık Uygulama Saati / Course Hours For Week (Objected)	0.00	
Haftalık Laboratuar Saati / Course Hours For Week (Laboratory)	0.00	
Dersin Verildiği Yıl / Year	1	
Öğretim Sistemi / Teaching System	Face to Face / Face to Face	
Eğitim Dili / Education Language		
Ön Koşulu Olan Ders(ler) / Precondition Courses	Yok	None
Amacı / Purpose	Dersin amacı modern yazılım mühendisliğinde ileri teknolojiyi destekleyen çeşitli çağdaş konular üzerinde çalışmaktır.	The aim of this course is to study various contemporary and advanced topics in modern software engineering which supports state of the art and technology.
İçeriği / Content	Dersin içeriği şu konular üzerinde yoğunlaşmaktadır: Yazılım sınaması ve doğrulama, büyük ölçekli yazılım mühendisliği, gömülü ve IoT yazılımlara giriş, yazılım mühendisliğinde araştırma metodolojileri, yazılım sürüm kontrolü ve yeniden yapılandırma, SR ve SLR.	The content of the course focuses on: Software V&V, Large Scale Software Engineering, Introduction to Embedded and IoT software, Research Methodologies in Software Engineering, Software Version Control and Refactoring, Systematic Review and SLR.
Önerilen Diğer Hususlar / Recommended Other Considerations	Bu dersi almadan önce 'Nesne Yönelimli Programlama' ve 'Programlama Dilleri' hakkında bilgi sahibi olmanız öneriliyor.	Nonelt is recommended to have information about the concepts of Object-oriented Programming and Programming Languages prior to this course.
Staj Durumu / Internship Status	Yok	None
Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	<ul style="list-style-type: none"> <li>• Pierre Bourque and Richard E. (Dick) Fairley "Software Engineering Body of Knowledge", Version 3, IEEE Computer Society Press, 2014.</li> <li>• Daniel Jackson, "Software Abstractions - Logic, Language, and Analysis", Revised Edition, MIT Press, 2012.</li> <li>• Sommerville, I., "Software Engineering", 10th Edition, Pearson, 2016.</li> </ul>	<ul style="list-style-type: none"> <li>• Pierre Bourque and Richard E. (Dick) Fairley "Software Engineering Body of Knowledge", Version 3, IEEE Computer Society Press, 2014.</li> <li>• Daniel Jackson, "Software Abstractions - Logic, Language, and Analysis", Revised Edition, MIT Press, 2012.</li> <li>• Sommerville, I., "Software Engineering", 10th Edition, Pearson, 2016.</li> <li>• Pressman, R.S., "Software Engineering: A Practitioner's Approach", 8th Edition, McGraw Hill. 2015.</li> </ul>
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)	Asst. Prof. Dr. Moharram Challenger	

## ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	1. Yazılım sına ve doğrulama konusunda bilgi sahibi olma	1. Having knowledge of software Validation & Verification
2	2. Büyük ölçekli yazılım mühendisliği yapabilme yeteneđi	2. Ability to perform Large Scale Software Engineering
3	3. Gömülü ve IoT yazılım geliştirme becerisi	3. Ability to embedded and IoT software development
4	4. Yazılım mühendisliğinde araştırma yöntemleri bilgisine sahip olmak	4. Having Knowledge of Research Methodologies in Software Engineering
5	5. Yazılım sürümü kontrolü ve yeniden yapılandırma yeteneđi	5. Ability to perform Software Version Control and Refactoring
6	6. “Systematic Review (SR)” ve “Systematic Literature Review (SLR)” yapabilme yeteneđi	6.Ability to conduct Systematic Review and Systematic Literature Review

## HAFTALIK DERS İÇERİĐİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım geliştirme metodolojileri	Okuma			
	Software Development Methodologies	Reading			
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	UML tabanlı yazılım tasarımı	Ödev			
	UML based software Design	Software design assignement			
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	SAT çözücülerini kullanarak yazılım sına ve doğrulama	Internet araştırması (durum çalışması)			
	Software Validation and Verification using SAT solvers	Internet research			
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Buyuk ölçek'te sistemler için yazılım mühendisliği	Okuma			
	Software Engineering of Large Scale Systems	Reading			
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Büyük ölçekli yazılım geliştirme için mevcut metodolojileri değiştirmek, genişletmek ve/veya birleştirmek için yaklaşımlar	Araştırma raporu			
	Approaches for modifying, extending and/or combining existing methodologies for large scale software development	Preparing research report			

	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
6	Gömülü sistemler için yazılım geliştirme	Okuma			
	Embedded Software Development	Reading			
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	IoT için yazılım geliştirme	Okuma			
	IoT Software Development	Reading			
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Ara Sınav				
	Midterm				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Çoklu paradigma modellemesi: Yazılım ve fiziksel sistemlerin kombinasyonu	Okuma			
	Multi Paradigm Modeling: Combination of Software and Physical Systems	Reading			
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım mühendisliğinde araştırma metodolojisi	İnternet araştırması (durum çalışması)			
	Research Methodology in Software Engineering	Internet research			
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Sürüm kontrolü ve yeniden yapılandırma	İnternet araştırması (durum çalışması)			
	Version Control and Refactoring	Internet research			

	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
12	Yazılım mühendisliğinde tasarım paternleri	Ödev			
	Patterns in Software Engineering	Design pattern assignment			
13	Yazılım mühendisliğinde "Systematic Review" ve "Systematic Literature Review"	Araştırma raporu			
	Systematic Review and Systematic Literature Review in Software Engineering	Preparing research report			
14	Yazılım mühendisliğine çağdaş yaklaşımlar: Üretken yazılım mühendisliği	Ödev			
	Contemporary Approaches to Software Engineering: Generative Software Engineering	Generative software development assignment			
15	Donem proje sunumu	Sunum			
	Project presentation	Presentation			
16	Final Sınav				
	Final Exam				

## DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçi Etkinlikleri / Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Ara Sınav / Midterm Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		40

  

Yarıyıl (Yıl) Sonu Etkinlikleri / End Of Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Final Sınavı / Final Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		60

  

Etkinliklerinin Başarı Notuna Katkı Yüzdesi(%) Toplamı / Total Percentage of Contribution (%) to Success Grade:	100
Değerlendirme Tipi / Evaluation Type:	

İŞ YÜKÜ / WORKLOADS

Etkinlikler / Workloads	Sayı / Number	Süresi (Saat) / Duration (Hours)	Toplam İş Yüğü (Saat) / Total Work Load (Hour)
Ara Sınav / Midterm Examination	1	38.00	38.00
Final Sınavı / Final Examination	1	53.00	53.00
Derse Katılım / Attending Lectures	14	3.00	42.00
Takım/Grup Çalışması / Team/Group Work	3	3.00	9.00
Rapor Hazırlama / Report Preparation	2	7.00	14.00
Proje Hazırlama / Project Preparation	1	50.00	50.00
Proje Sunma / Project Presentation	1	10.00	10.00
Ödev Problemleri için Bireysel Çalışma / Individual Study for Homework Problems	3	4.00	12.00
Okuma / Reading	5	4.00	20.00
<b>Toplam / Total:</b>	<b>31</b>	<b>172.00</b>	<b>248.00</b>
<p>Dersin AKTS Kredisi = Toplam İş Yüğü (Saat) / 30.00 (Saat/AKTS) = 248.00/30.00 = 8.27 ~ / Course ECTS Credit = Total Workload (Hour) / 30.00 (Hour / ECTS) = 248.00 / 30.00 = 8.27 ~</p>			

PROGRAM VE ÖĞRENME ÇIKTISI / PROGRAM LEARNING OUTCOMES

Öğrenme Çıktıları / Learning Outcomes	Program Çıktıları / Program						
	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7
1.1. Yazılım sınama ve doğrulama konusunda bilgi sahibi olma / 1. Having knowledge of software Validation & Verification							
2.2. Büyük ölçekli yazılım mühendisliği yapabilme yeteneği / 2. Ability to perform Large Scale Software Engineering							
3.3. Gömülü ve IoT yazılım geliştirme becerisi / 3. Ability to embedded and IoT software development							
4.4. Yazılım mühendisliğinde araştırma yöntemleri bilgisine sahip olmak / 4. Having Knowledge of Research Methodologies in Software Engineering							
5.5. Yazılım sürümü kontrolü ve yeniden yapılandırma yeteneği / 5. Ability to perform Software Version Control and Refactoring							
6.6. "Systematic Review (SR)" ve "Systematic Literature Review (SLR)" yapabilme yeteneği / 6.Ability to conduct Systematic Review and Systematic Literature Review							

Katkı Düzeyi / Contribution Level : 1-Çok Düşük / Very low, 2-Düşük / Low, 3-Orta / Moderate, 4-Yüksek / High, 5-Çok Yüksek / Very high